

***Examiner's Amendment***

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Ted Liu on 8/11/2009.

1. (Currently Amended) A method for navigating a graphical user interface (GUI) having at least one page, comprising:

providing a first booklet, wherein user interaction with the first booklet can cause the GUI to navigate to a new page;

providing a request based on user interaction with the first booklet;

mapping the request to a control tree factory;

generating a control tree from the factory based on the request wherein the control tree includes a booklet control corresponding to the first booklet;

advancing the control tree through at least one lifecycle stage based on the request;

creating a metadata representation of a control tree;

generating a class to construct the control tree based on the metadata representation;

and

generating a response wherein the response can be used to render the new page;

wherein the at least one lifecycle stage includes an event lifecycle stage where at least one control of the control tree raise events to communicate with another control of the control tree; wherein the event stage occurs before a render lifecycle stage and wherein in the render stage the controls of the control tree create their own GUI representation; and

wherein a pre-render lifecycle stage occurs between the event lifecycle stage and the render lifecycle stage and wherein an additional control is dynamically added to the control tree at a stage before the pre-render lifecycle stage and wherein when the additional control is added to the control tree dynamically, a lifecycle catch-up process drives the additional control through lifecycle stages until the additional control catches-up to a current stage;

wherein when current stage is at a pre-render stage or later, the additional control is driven through at least an "init", "load" and "raise events" lifecycle stage so that the additional control catches up to the current state;

wherein the controls of the control tree include a number of classes including a base control class; the base control class defines a set of services available to all controls,

wherein the lifecycles are defined by a set of methods representing stages in the lifecycle, a control can override methods of the lifecycle, at least one lifecycle stage can comprise two methods that are called by lifecycle driver;

wherein a Lifecycle class defines methods called during the control's lifecycle;  
lifecycle events can have two methods, a base method to provide the lifecycle service  
for the control, and a method called to raise an event;

wherein the controls have three types of names:

a user assigned value that is unique within a name scope,

a generated name that is unique within a scope, and

a globally unique name of the control;

wherein the naming scope provides support for

uniquely naming a control within a tree;

unique names are used for state management.

5. (Canceled).

16. (Currently Amended) A method for navigating a portal graphical user  
interface (GUI) having at least one page, comprising:

providing a first booklet, wherein user interaction with the first booklet can cause  
the GUI to navigate to a new portal page;

providing a request based on user interaction with the first booklet;

mapping the request to a control tree factory;

generating a control tree from the factory based on the request wherein the  
control tree includes a booklet control corresponding to the first booklet;

advancing the control tree through at least one lifecycle stage based on the request;

creating a metadata representation of a control tree;

generating a class to construct the control tree based on the metadata representation;

generating a response wherein the response can be used to render the new portal page; and

wherein the new page can be a second booklet;

wherein the at least one lifecycle stage includes an event lifecycle stage where at least one control of the control tree raise events to communicate with another control of the control tree; wherein the event lifecycle stage occurs before a render lifecycle stage and wherein in the render stage, the controls of the control tree create their own GUI representation; and

wherein a pre-render lifecycle stage occurs between the event lifecycle stage and the render lifecycle stage; and

wherein an additional control is dynamically added to the control tree at a stage before the pre-render lifecycle stage; and

wherein when the additional control is added to the control tree dynamically, a lifecycle catch-up process drives the additional control through lifecycle stages until the additional control catches-up to a current stage;

wherein when current stage is at a pre-render stage or later, the additional control is driven through at least an “init”, “load” and “raise events” lifecycle stage so that the additional control catches up to the current state;

wherein the controls of the control tree include a number of classes including a base control class; the base control class defines a set of services available to all controls,

wherein the lifecycles are defined by a set of methods representing stages in the lifecycle, a control can override methods of the lifecycle, at least one lifecycle stage can comprise two methods that are called by lifecycle driver;

wherein a Lifecycle class defines methods called during the control's lifecycle; lifecycle events can have two methods, a base method to provide the lifecycle service for the control, and a method called to raise an event;

wherein the controls have three types of names:

a user assigned value that is unique within a name scope,

a generated name that is unique within a scope, and

a globally unique name of the control;

wherein the naming scope provides support for

uniquely naming a control within a tree;

unique names are used for state management.

30. (Currently Amended) A machine readable storage medium having instructions stored thereon that when executed by a processor cause a system to:

provide a first booklet, wherein user interaction with the first booklet can cause a graphical user interface (GUI) to navigate to a new page;

provide a request based on user interaction with the first booklet;

map the request to a control tree factory;

generate a control tree from the factory based on the request wherein the control tree includes a booklet control corresponding to the first booklet;

advance the control tree through at least one lifecycle stages based on the request;

create a metadata representation of a control tree;

generate a class to construct the control tree based on the metadata representation;

and

generate a response wherein the response can be used to render the new page; wherein the at least one lifecycle stage includes an event lifecycle stage where at least one control of the control tree raise events to communicate with another control of the control tree; wherein the event lifecycle stage occurs before a render lifecycle stage and wherein in the render stage, the controls of the control tree create their own GUI representation; and

wherein a pre-render lifecycle stage occurs between the event lifecycle stage and the render lifecycle stage and wherein an additional control is dynamically added to

the control tree at a stage before the pre-render lifecycle stage and wherein when the additional control is added to the control tree dynamically, a lifecycle catch-up process drives the additional control through lifecycle stages until the additional control catches-up to a current stage;

wherein when current stage is at a pre-render stage or later, the additional control is driven through at least an "init", "load" and "raise events" lifecycle stage so that the additional control catches up to the current state;

wherein the controls of the control tree include a number of classes including a base control class; the base control class defines a set of services available to all controls,

wherein the lifecycles are defined by a set of methods representing stages in the lifecycle, a control can override methods of the lifecycle, at least one lifecycle stage can comprise two methods that are called by lifecycle driver;

wherein a Lifecycle class defines methods called during the control's lifecycle; lifecycle events can have two methods, a base method to provide the lifecycle service for the control, and a method called to raise an event;

wherein the controls have three types of names:

a user assigned value that is unique within a name scope,

a generated name that is unique within a scope, and

a globally unique name of the control;

wherein the naming scope provides support for

uniquely naming a control within a tree;

unique names are used for state management.

31. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the first booklet is at least one of: 1) a set of tabs and/or buttons; and 2) a menu.

32. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the first booklet is associated with at least one of the least one page.

33. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the new page can a second booklet.

34. (Cancelled.)

35. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the request is an hypertext transfer protocol request (HTTP); and

the request originates from a web browser.



36. (Currently Amended) The machine readable storage medium of claim 30, further comprising instructions that when executed cause the system to:  
providing the response to a web browser.

37. (Currently Amended) The machine readable storage medium of claim 30 wherein:  
the control tree is driven through the at least one lifecycle stage by an interchangeable lifecycle component.

38. (Currently Amended) The machine readable storage medium of claim 30 wherein:  
the booklet control has an interchangeable persistence mechanism.

39. (Currently Amended) The machine readable storage medium of claim 30 wherein:  
the booklet control can render itself according to a theme.

40. (Currently Amended) The machine readable storage medium of claim 30 wherein:  
the booklet control can interact with another of the at least one controls.

Art Unit: 2174

41. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the booklet control can advance through the at least one lifecycle stage in parallel with other controls in the control tree.

42. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the at least one lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

43. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the response is an hypertext transfer protocol (HTTP) response.

44. (Currently Amended) The machine readable storage medium of claim 30 wherein:

the booklet control can raise events and respond to events.

45. (Currently Amended) A computer readable storage medium, comprising:

a code segment including instructions to provide a first booklet, wherein user interaction with the first booklet can cause a graphical user interface (GUI) to navigate to a new page;

a code segment including instructions to provide a request based on user interaction with the first booklet;

a code segment including instructions to map the request to a control tree factory;

a code segment including instructions to generate a control tree from the factory based on the request wherein the control tree includes a booklet control corresponding to the first booklet;

a code segment including instructions to advance the control tree through at least one lifecycle stage based on the request;

a code segment including instructions to create a metadata representation of a control tree;

a code segment including instructions to generate a class to construct the control tree based on the metadata representation; and

a code segment including instructions to generate a response wherein the response can be used to render the new page;  
wherein the at least one lifecycle stage includes an event lifecycle stage where at least one control of the control tree raise events to communicate with another control of the control tree; wherein the event lifecycle stage occurs before a render lifecycle stage and wherein in the render stage, the controls of the control tree create their own GUI representation; and

wherein a pre-render lifecycle stage occurs between the event lifecycle stage and the render lifecycle stage and wherein an additional control is dynamically added to

the control tree at a stage before the pre-render lifecycle stage and wherein when the additional control is added to the control tree dynamically, a lifecycle catch-up process drives the additional control through lifecycle stages until the additional control catches-up to a current stage;

wherein when current stage is at a pre-render stage or later, the additional control is driven through at least an "init", "load" and "raise events" lifecycle stage so that the additional control catches up to the current state;

wherein the controls of the control tree include a number of classes including a base control class; the base control class defines a set of services available to all controls,

wherein the lifecycles are defined by a set of methods representing stages in the lifecycle, a control can override methods of the lifecycle, at least one lifecycle stage can comprise two methods that are called by lifecycle driver;

wherein a Lifecycle class defines methods called during the control's lifecycle; lifecycle events can have two methods, a base method to provide the lifecycle service for the control, and a method called to raise an event;

wherein the controls have three types of names:

a user assigned value that is unique within a name scope,

a generated name that is unique within a scope, and

a globally unique name of the control;

wherein the naming scope provides support for

uniquely naming a control within a tree;

unique names are used for state management.

46. (Canceled).

47. (Currently Amended) The method of claim [[46]] 1, wherein a Context class provides services to controls in the tree, these services are protocol-independent; the context provides the following services: 1) Access to an Abstract Data Layer, the context class providing an object implementing the interface providing an abstract data layer, 2) Request Type, the context class contains properties indicating what type of request is happening, 3) Request Information, the context class provides information that comes from the request; and 4) Generic Services, the context class allows controls to access and register as service providers to other controls.

48. (Currently Amended) The method of claim 47 wherein the Renderer class is an interface that allows a class to be created and which acts as a rendering proxy for a control; a control has rendering methods allowing the control to output its representation into a stream; wherein there are two primary methods for rendering: beginRender( ) and endRender( ); wherein Default rendering can be overridden by setting a control's renderer by providing an object implementing the Renderer interface.

***Reasons for Allowance***

1. Amendment to include the limitations from the cancelled claims 5 and 46 into Independent claims of 1, 16, 30, and 45 specifies methods, machine readable storage medium and a computer readable storage medium, which now further require:

*a. creating a metadata representation of a control tree; generating a class to construct the control tree based on the metadata representation;*

*b. wherein the controls have three types of names:*

*a user assigned value that is unique within a name scope,*

*a generated name that is unique within a scope, and*

*a globally unique name of the control;*

*wherein the naming scope provides support for*

*uniquely naming a control within a tree;*

*unique names are used for state management.*

2. The amendment of independent claims 1 (method), 16 (method), 30 (machine readable storage medium), and 45 (computer readable storage medium) combines the above recited limitations, along with the features already present in the claims, in particular:

*wherein the controls of the control tree include a number of classes including a base control class; the base control class defines a set of services available to all controls,*

*wherein the lifecycles are defined by a set of methods representing stages in the lifecycle, a control can override methods of the lifecycle, at least one lifecycle stage can comprise two methods that are called by lifecycle driver;*

*wherein a Lifecycle class defines methods called during the control's lifecycle; lifecycle events can have two methods, a base method to provide the lifecycle service for the control, and a method called to raise an event;*

3. The combined features, both amended and already present, overcome the cited references of Hunter and Popp, and are not found in the art. Accordingly, each of the dependent claims are allowable by virtue of their dependency upon independent claims 1, 16, 30, and 45.

### **Conclusion**

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andrew Belousov whose telephone number is (571) 270-1695. The examiner can normally be reached on Mon-Fri (alternate Fri off) EST. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dennis Chow can be reached on (571) 272-7767. The fax phone number for the organization where this application or proceeding is assigned is 571-273-3800.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Steven P Sax/

Primary Examiner, Art Unit 2174

AB

8/15/2009